09-2407

PTO/SB/21 (04-07)
Approved for use through 09/30/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission	28

Application Number	09/841,503
Filing Date	04/24/2001
First Named Inventor	Richard Alan Dayan
Art Unit	2131
Examiner Name	Matthew T. Henning
Attorney Docket Number	RPS9 20010011

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☑ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) (please Identify below):

Remarks

Sent by Express Mail, No. EB034087403US

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Ronald V. Davidge, Inc.		
Signature	Ronald V. Davidge		
Printed name	Ronald V. Davidge		
Date	09/21/2007	Reg. No.	33,863

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Ronald V. Davidge		
Typed or printed name	Ronald V. Davidge	Date	07/21/2007

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Richard Alan Dayan, et al.
Application No. 09/841,503
Filed: 04-24-2001
For: SECURE SYSTEM AND METHOD FOR UPDATING A PROTECTED PARTITION
Group Art Unit: 2131
Examiner: Matthew T. Henning

BRIEF ON APPEAL

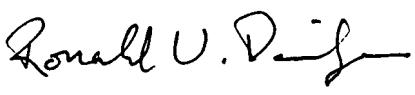
Honorable Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This document is the Appellant's brief in the above described application, for which a Notice of Appeal has been filed on 07/21/2007.

The Director is hereby authorized to charge a fee of $500.00 for filing an appeal brief, and to charge any additional fees which may be required, and to credit any overpayment, to Deposit Account No. 50-3533. I have enclosed a duplicate copy of this sheet.

Respectfully submitted by:



Ronald V. Davidge
Registration No. 33,863
Telephone No. 954-736-0203

TABLE OF CONTENTS

Real Party of Interest

The real party of interest is Lenovo (Singapore) Pte Ltd.

Related Appeals and Interferences

There are no related appeals or interferences.

Status of Claims

Claims 37-62 are pending. Claims 1-36 have been canceled. No claims have been allowed, and there are no other claims pending. Claims 37, 38, 44-52, and 57-62 are the subject of this appeal, with claims 39-43 and 53-56 having been withdrawn from the appeal.

Status of Amendments

An amendment to the application has been filed on 09/12/2007, after filing the Notice of Appeal. In this amendment, claims 37-43 and 50-56 are canceled; claim 44 is rewritten in independent form to include all the limitations of claims 37 and 38, upon which this claim formerly depended; and claim 57 is rewritten in independent form to include all the limitations of claims 50 and 51, upon which this claim formerly depended. This amendment has been received by the Patent and Trademark Office, but there is presently no indication that this amendment has been considered by the Examiner, or that it has been entered or rejected.

Summary of the Claimed Subject Matter

Claim 37

According to claim 37, a method for providing a capability to securely update information stored in a plurality of computer systems 10, such as the computer system 10, shown in FIG. 1, is provided. (It is noted that, in FIG. 1 as originally submitted, only one computer system 10 was shown. However, in an amendment filed on 08/04/2006, this drawing was amended to show a plurality of computer systems 10. Support for this amendment to FIG. 1 is found in the specification as originally filed on page 11, lines 11-14).

The method comprises:

forming a protected partition 38 within a hard drive 18 of each of the computer systems 10 (as shown in FIG. 1);

storing, within nonvolatile storage 42, 22, 18 of each computer system 10 (shown in FIG. 1) in the plurality of computer systems 10, a setup password 44, an operating system 40, and an initialization routine 36 (as described on page 10, lines 14-19) to execute within a processor 12 of the computer system 10 after power on of the computer system 10, wherein the initialization routine 36 includes instructions causing the protected partition 38 to be locked (as in step 198 of FIG. 6A, described on page 21, lines 1-5) before the operating system 40 is loaded (as described on page 10, lines 21-22, and as in step 200 of FIG. 6A, described on page 21, lines 5-6), and wherein instructions causing information stored within a predetermined location to be written within the protected partition 38 after predetermined security procedures using the setup password 44 have occurred (in step 196 of FIG. 6A, described on page 21, lines 7-9) but before the protected partition 38 is locked (again, in step 198 of FIG. 6A);

establishing a network 14, connecting each computer system 10 in the plurality of computer systems 10 with a server system 11, (as descried on page 11, lines 11-14);

generating an update partition file within the server system 11 (as shown in FIG. 5 and described on page 13, line 3, through page 14, line 24);

transmitting the update partition file over the network 34 to each computer system 10 in the plurality of computer systems 10 (as described in page 15, lines 1-6); and

storing the update partition file 56 (shown in FIGS. 2 and 3) within the predetermined location of each computer system 10 in the plurality of computer systems 10 (as described in page 15, lines 6-9).

Claim 38

According to this claim, the method of claim 37 is provided, wherein the initialization routine 36 (shown in FIG. 1) includes instructions causing the processor 12 of the computer system 10 to perform a method (shown in FIG. 6A and 6B) including:

comparing information stored in the protected partition 38 with information from the update partition file (of FIGS. 2, 3) stored within the predetermined location (in steps 162, 164, and 168 of FIGS. 6A, 6B, described on page 18, line 21, through page 19, line 5);

when a portion of the information stored in the protected partition 38 is found to match a portion of the information stored within the update partition file 56, overwriting the portion of the information stored in the protected partition 38 with the portion of the information stored in the update partition file 56 if space around the portion of the information stored in the protected partition is sufficient (in steps 170, 172, and 174 of FIG. 6B, described on page 19, lines 5-16);

when a portion of the information stored in the protected partition 38 is not found to match a portion of the information stored within the update partition file 56, writing the portion of the information stored within the update partition file 56 to append to the information stored in the protected partition 38 if space within the protected partition 38 is sufficient (in steps 182 and 184 of FIG. 6B, described on page 19, lines 10-16); and

locking the protected partition 38 to prevent further modification of information stored within the protected partition 38 (in step 198 of FIG. 6A, described on page 21, lines 3-5).

Claim 44

According to this claim, the method of claim 38 is provided, wherein

the update partition file 56 (shown in FIGS. 2, 3) includes a plurality of entries 60 and a plurality of encrypted elements 64 (which has been encrypted, for example, using the private key of the server 11, as described on page 17, lines 24-25),

each entry within the plurality of entries includes information to be stored at a different location within the protected partition (as described on page 12, lines 14-15),

each encrypted element 64 within the plurality of encrypted elements 64 is associated with an entry 60 in the plurality of entries 60 (as described on page 12. line 21, through page 13, line 2),

the method additionally comprises, following determining that the update partition file 56 is stored within the computing system 10 for updating the protected partition 38 (for example, by determining in step 124 of FIG. 6A that the protected partition flag has been set, as described on page 17, lines 8-10), verifying whether each entry 60 in the plurality of entries 60 within the update partition file 56 has been generated by the server system 11, and

each entry 60 in the plurality of entries 60 within the update partition file 56 is written to the protected partition 38 only following verification that the entry 60 has been generated by the server system 11 (using the AUTHENTICATE program of FIG. 7 as described on page 17, line 10 through page 11, line 11, and further as described in steps 150, 152, and 153 of FIG. 6A, as described on page 18, lines 12-20. Also note that this is done for each entry, as a determination is made in step 120 of FIG. 6A of whether additional entries are present in the update partition and repeating the process before resetting the update flag in step 183, as described in page 20, lines 13-16).

Claim 50

According to this claim, an interconnected system (shown in FIG. 1) is provided for providing updated information in a secure manner, wherein the interconnected system comprises:

a network 34 (as described on page 10, lines 8-11);

a server system 11 connected to the network 34 and programmed to generate an update partition file 56 (with the file 56 being shown in FIGS. 2 and 3, and as described on page 12, line 4, through page 13, line 2) and to transmit the update partition file 56 over the network (as described on page 11, lines 23-25, and further with the process of generating the update partition file as shown in FIG. 5, being described on page 13, line 3, through page 16, line 24); and

a computer system 10 connected to the network 34, wherein the computer system 10 (as described on page 10, lines 1-13) includes a processor 12, non-volatile data storage 42, 22, 18

including a hard drive 18 having a protected partition 38, wherein the processor 12 is programmed to receive the update partition file 56 from the network 34 and to store the update partition file 56 in a predetermined location within the nonvolatile data storage 42, 18 outside the protected partition 38, and wherein the nonvolatile data storage 42, 22, 18 stores an operating system 40 and an initialization routine 36, executing within the processor 12 after power on of the computer system 10, including instructions causing the protected partition 38 to be locked (in step 198 of FIG. 6A, and as described on page 21, lines 3-50 before the operating system 40 is loaded, and instructions causing information stored within the predetermined location to be written within the protected partition 38 after predetermined security procedures (in step 196 of FIG. 6A, described on page 21, lines 7-9) have occurred but before the protected partition 38 is locked (again, in step 198 of FIG. 6A).

Claim 51

According to this claim, the interconnected system of claim 50 is provided, wherein the initialization routine 36 (shown in FIG. 1) includes instructions causing the processor 12 of the computer system 10 to perform a method (shown in FIG. 6A and 6B) including:

comparing information stored in the protected partition 38 with information from the update partition file (of FIGS. 2, 3) stored within the predetermined location (in steps 162, 164, and 168 of FIGS. 6A, 6B, described on page 18, line 21, through page 19, line 5);

when a portion of the information stored in the protected partition 38 is found to match a portion of the information stored within the update partition file 56, overwriting the portion of the information stored in the protected partition 38 with the portion of the information stored in the update partition file 56 if space around the portion of the information stored in the protected partition is sufficient (in steps 170, 172, and 174 of FIG. 6B, described on page 19, lines 5-16);

when a portion of the information stored in the protected partition 38 is not found to match a portion of the information stored within the update partition file 56, writing the portion of the information stored within the update partition file 56 to append to the information stored in the protected partition 38 if space within the protected partition 38 is sufficient (in steps 182 and 184 of FIG. 6B, described on page 19, lines 10-16); and

locking the protected partition 38 to prevent further modification of information stored within the protected partition 38 (in step 198 of FIG. 6A, described on page 21, lines 3-5).

Claim 57

According to this claim, the interconnected system of claim 51 is provided, wherein

the update partition file 56 (shown in FIGS. 2, 3) includes a plurality of entries 60 and a plurality of encrypted elements 64 (which has been encrypted, for example, using the private key of the server 11, as described on page 17, lines 24-25),

each entry within the plurality of entries includes information to be stored at a different location within the protected partition (as described on page 12, lines 14-15),

each encrypted element 64 within the plurality of encrypted elements 64 is associated with an entry 60 in the plurality of entries 60 (as described on page 12. line 21, through page 13, line 2),

the method additionally comprises, following determining that the update partition file 56 is stored within the computing system 10 for updating the protected partition 38 (for example, by determining in step 124 of FIG. 6A that the protected partition flag has been set, as described on page 17, lines 8-10), verifying whether each entry 60 in the plurality of entries 60 within the update partition file 56 has been generated by the server system 11, and

each entry 60 in the plurality of entries 60 within the update partition file 56 is written to the protected partition 38 only following verification that the entry 60 has been generated by the server system 11 (using the AUTHENTICATE program of FIG. 7 as described on page 17, line 10 through page 11, line 11, and further as described in steps 150, 152, and 153 of FIG. 6A, as described on page 18, lines 12-20. Also note that this is done for each entry, as a determination is made in step 120 of FIG. 6A of whether additional entries are present in the update partition and repeating the process before resetting the update flag in step 183, as described in page 20, lines 13-16).

Grounds of Rejection to be Reviewed on Appeal

1. Whether Claims 44-49 and 57-62 are Unpatentable under 35 USC §103(a) over U.S. Patent Number 6,026,016 to Gafken in view of U.S. Patent Number 5,128,995 to Arnold et al., further in view of "Handbook of Applied Cryptography" by Menezes et al., further in view of U.S. Patent. Number 6,088,759 to Hasbun, et al., and further in view of U.S. Patent Application Number 2001/0039561 A1 to Hayashi, et al.

Argument

1. Whether Claims 44-49 and 57-62 are Unpatentable under 35 USC §103(a) over U.S. Patent Number 6,026,016 to Gafken in view of U.S. Patent Number 5,128,995 to Arnold et al., further in view of "Handbook of Applied Cryptography" by Menezes et al., further in view of U.S. Patent Number 6,088,759 to Hasbun, et al., and further in view of U.S. Patent Application Number 2001/0039561 A1 to Hayashi, et al.

The Appellant submits that *Gafken, Arnold et al., Menezes et al.,. Hasbun et al.* and *Hayashi et al.,* taken separately or in combination fail to teach, describe or otherwise anticipate the requirements of claims 44 and 57 for the update partition file to include a plurality of entries and a plurality of encrypted elements, wherein each encrypted element in the plurality of encrypted elements is associated with an entry in the plurality of entries, wherein the method performed by the initialization program includes verifying whether each entry In the plurality of entries has been generated by the server system, and wherein each entry in the plurality of entries is written to the protected partition only following verification that the entry has been generated by the server system.

The method for updating code within the apparatus of *Gafken* is shown in FIG. 5 therein. with a validation process being shown in FIG. 6 and described in column 12, line 12, through column 13, line 5. The validation process of *Gafken* is described as including a step 605 of verifying that a server on a network with which the computer system communicates is a valid computer system from which to download the code image, for example by performing a challenge using passwords or keys, a step 610 of verifying that the vendor of the code image matches the vendor of the existing code, and a step 615 of ensuring that the code image has not been tampered with, including, for example, recomputation of a digest of the code image and comparison ith a digest value of the code image signature. *Gafken* further describes a process for updating the stored image in steps 550, 555, 560, and 565 of FIG. 5, with an optional validation process occurring in step 560, and is described in column 13, line 52, through column 14, line 10. There is no

indication within *Gafken* that a separate validation process performed for each of multiple entries within the code image; the entire image is validated as a whole.

In the Office Action of 03/21/2007, the Examiner described *Arnold et al.* as teaching that a BIOS can be stored in a protected partition of a hard drive. Since *Arnold et al.* does not discuss updating code stored within the protected partition from a remote server, adding the teachings of *Arnold et al.* to the description of *Gafken* does not overcome the deficiencies described above of *Gafken* in describing the limitations of claims 44 and 57. The Examiner further described *Menezes et al.* only as teaching that providing a sequence number, or password, stored and updated at both a receiver and a sender, in a digital signature of the sender, protects the signature against replay attacks. Adding this teaching of *Menezes et al.* to the description of *Gafken* does not overcome the deficiencies described above of *Gafken* in describing the limitations of claims 44 and 57. The Examiner additionally described *Hasbun et al.* only as teaching that a BIOS update can be allocated into virtual blocks to that the blocks can be updated individually without having to erase the entire memory first, and that new blocks should be allocated from existing free memory. Adding these teachings of *Hasbun et al.* does not overcome the deficiencies described above of *Gafken* in describing the limitations of claims 44 and 57. The Examiner further said that a combination of *Gafken et al., Arnold et al., Menezes et al.,* and *Hasbun et al.* failed to disclose encrypting portions of the file separately and verifying each portion individually. The Examiner additionally said that *Hayashi et al.* teaches a method for providing a variety of software safely by breaking the file into pieces and decrypting each piece separately.

However, the Appellant notes that *Hayashi et al.* merely describes a file including differently encrypted entries, not a file containing a plurality of entries and a plurality of encrypted elements, with each of the encrypted elements associated with one of the entries, as required by claims 44 and 57. In addition, it is noted that *Hayashi et al.* does not describe a process for verifying that individual portions of the stored data, as required by the claims.

Furthermore, in the Office Action of 03/21/2007, the Examiner said that the Appellantm's claims do not recite that the source of each entry is verified independently for each entry. The Examiner further said that the signature verification of *Gafken et al.,* which verifies the source of the entire

upgrade, and therefore of each entry, through digital signatures, meets the limitations of claims 44 and 57.

The Appellant submits that the above statement by the Examiner inherently includes an assumption that all of the entries in an update partition file stored within the computer system to be written to the protected partition have been written by the same source. In an important scenario, the update partition file may include a damaging file, such as a newly devised virus, sent from a computer system other than the trusted server, and a legitimate file, sent from the trusted server, such as a software update designed to protect the computer system from the newly devised virus. Such a scenario would be expected to occur following the discovery of a new virus being spread among computer systems. It is noted that, in the Appellant's invention, entries within the update partition file are written to the protected partition only during an initialization routine executed at startup of the computer system, with a substantial time generally elapsing with power on to the computer system between system startup sequences. In this way, plenty of time is provided for a number of different entries to be written to the update partition file from the trusted server and from one or more other computer systems. The method described in claims 44 and 57 is thus understood to be not only different from that of *Gafken et al.*, but superior thereto, in that the separate verification of elements within the update partition file makes it possible for entries that have been verified to be written to the protected partition while entries that have not been verified are not written to the protected partition.

In other words, given the fact that it is possible for entries within the update partition file to come from multiple sources, it is not possible to verify that each entry has been received from the trusted server while writing individual entries to the protected partition, as required by claims 44 and 47, using the method of *Gafken et al.*, as suggested by the Examiner. This can only be done by separately verifying each entry.

Thus, it is submitted that the method of *Gafken et al.* is not the equivalent of the method of claims 44 and 57, in which each entry within the plurality of entries within the plurality of entries in the update partition file is verified as having been written by the trusted server, with no entry being written to the protected partition only following verification that the entry has been

generated by the trusted server system. The ability of the method of the Appellant to write entries determined to be from the trusted server to the protected partition while not writing entries determined not to be from the trusted server constitutes a substantial difference between the Appellant's method and the method of *Gafken.*

For all the above reasons, it is believed that claims 44 and 57 are patentable under 35 USC §103(a) over U.S. Patent Number 6,026,016 to Gafken in view of U.S. Patent Number 5,128,995 to Arnold et al., further in view of "Handbook of Applied Cryptography" by Menezes et al., further in view of U.S. Patent Number 6,088,759 to Hasbun, et al., and further in view of U.S. Patent Application Number 2001/0039561 A1 to Hayashi, et al.

Because dependent claims 45-49 and 58-62 merely add limitations to claims 44 and 57, respectively, these dependent claims are additionally believed to be patentable for reasons described above regarding claims 44 and 57.

CLAIMS APPENDIX

37 A method for providing a capability to securely update information stored in a plurality of computer systems, wherein the method comprises:

forming a protected partition within a hard drive of each of the computer systems

storing, within nonvolatile storage of each computer system in the plurality of computer systems, a setup password, an operating system, and an initialization routine to execute within a processor of the computer system after power on of the computer system, wherein the initialization routine includes instructions causing the protected partition to be locked before the operating system is loaded, and wherein instructions causing information stored within ~~the~~ a predetermined location to be written within the protected partition after predetermined security procedures using the setup password have occurred but before the protected partition is locked;

establishing a network connecting each computer system in the plurality of computer systems with a server system;

generating an update partition file within the server system;

transmitting the update partition file over the network to each computer system in the plurality of computer systems; and

storing the update partition file within the predetermined location of each computer system in the plurality of computer systems.

38 The method of claim 37, wherein the initialization routine includes instructions causing the processor of the computer system to perform a method including:

comparing information stored in the protected partition with information from the update partition file stored within the predetermined location;

when a portion of the information stored in the protected partition is found

partition is found to match a portion of the information stored within the update partition file, overwriting the portion of the information stored in the protected partition with the portion of the information stored in the update partition file if space around the portion of the information stored in the protected partition is sufficient;

when a portion of the information stored in the protected partition is not found to match a portion of the information stored within the update partition file, writing the portion of the information stored within the update partition file to append to the information stored in the protected partition if space within the protected partition is sufficient; and

locking the protected partition to prevent further modification of

information stored within the protected partition.

44. The method of claim 38, wherein

the update partition file includes a plurality of entries and a plurality of

encrypted elements,

each entry within the plurality of entries includes information to be stored at a different location within the protected partition,

each encrypted element within the plurality of encrypted elements is associated with an entry in the plurality of entries,

the method additionally comprises, following determining that the update partition file is stored within the computing system for updating the protected partition, verifying whether each entry in the plurality of entries within the update partition file has been generated by the server system, and

each entry in the plurality of entries within the update partition file is written to the protected partition only following verification that the entry has been generated by the server system.

45. The method of claim 44, wherein verifying that ~~the~~ each entry in the plurality of entries within the update partition file has been generated by the server system includes:

forming a first message digest by applying a hash algorithm to the entry;

forming a second message digest by signing the encrypted element associated with the entry using a public key of the server system; and;

determining that the first and second message digests are identical.

46. The method of claim 44, wherein verifying that ~~the~~ each entry in the plurality of entries within the update partition file has been generated by the server system includes signing the encrypted element associated with the entry with a public key of the server system, and the encrypted element of the update partition file has been prepared by signing, with the private

key of the server system, a result of the application of an algorithm to data including a version of the setup password accessed by the server system.

47. The method of claim 46, wherein

the data includes the version of the setup password appended to ~~a~~ the entry,

the algorithm is a hash algorithm generating a message digest, and

verifying that the entry has been generated by the server system includes applying the hash algorithm to the setup password stored within the computing system appended the entry to generate a first version of a message digest and comparing the first version of the message digest with a second version of the message digest obtained by signing the encrypted element.

48. The method of claim 44, wherein

information stored in the protected partition is compared to each entry in the plurality of entries within the update partition file,

when a portion of the information stored in the protected partition is found to match the entry, the portion of the information stored in the protected partition is overwritten with the entry if space around the portion of the information stored in the protected partition is sufficient, and

when a portion of the information stored in the protected partition is not found to match the entry, the entry is appended to the information stored in the protected partition if space within the protected partition is sufficient.

49. The method of claim ~~38~~ 48, wherein

the method additionally comprises receiving an input signal from a keyboard of the computing system and comparing the input signal with a signal corresponding to a setup password stored in non-volatile storage within the computing system, and

the protected partition is left unlocked if the input signal matches the signal corresponding to the setup password.

50. An interconnected system for providing updated information in a secure manner, wherein the interconnected system comprises:

a network;

a server system connected to the network and programmed to generate an update partition file and to transmit the update partition file over the network; and

a computer system connected to the network, wherein the computer system includes a processor, non-volatile data storage including a hard drive having a protected partition, wherein the processor is programmed to receive the update partition file from the network and to store the update partition file in a predetermined location within the nonvolatile data storage outside the protected partition, and wherein the nonvolatile data storage stores an operating system and an initialization routine, executing within the processor after power on of the computer system, including instructions causing the protected partition to be locked before the operating system is loaded, and instructions causing information stored within the predetermined location to be written within the protected partition after predetermined security procedures have occurred but

before the protected partition is locked.

51. The interconnected system of claim 50, wherein the initialization routine includes instructions causing the processor of the computer system to perform a method including:

comparing information stored in the protected partition with information from the update partition file stored within the predetermined location;

when a portion of the information stored in the protected partition is found to match a portion of the information stored within the update partition file, overwriting the portion of the information stored in the protected partition with the portion of the information stored in the protected partition if space around the portion of the information stored in the protected partition is sufficient;

when a portion of the information stored in the protected partition is not found to match a portion of the information stored within the update partition file, writing the portion of the information stored within the update partition file to append to the information stored in the protected partition if space within the protected partition is sufficient; and

locking the protected partition to prevent further modification of information stored within the protected partition.

52. The interconnected system of claim 51, wherein

a flag bit is set in non-volatile storage within the computing system when the update partition file is stored at a predetermined location in non-volatile storage within the computing system, and

determining whether the update partition file is stored within the computing system for updating the protected partition is performed by determining whether the flag bit is set.

57. The interconnected system of claim 51, wherein

the update partition file includes a plurality of entries and a plurality of encrypted elements,

each entry within the plurality of entries includes information to be stored at a different location within the protected partition,

each encrypted element within the plurality of encrypted elements is associated with an entry in the plurality of entries.

the method additionally comprises, following determining that the update partition file is stored within the computing system for updating the protected partition, verifying whether each entry in the plurality of entries within the update partition file has been generated by the server system, and

each entry in the plurality of entries within the update partition file is written to the protected partition only following verification that the entry has been generated by the server system.

58. The interconnected system of claim 57, wherein verifying that ~~the~~ each entry in the plurality of entries within the update partition file has been generated by the server system includes:

forming a first message digest by applying a hash algorithm to the entry;

forming a second message digest by signing the encrypted element associated with the entry using a public key of the server system; and;

determining that the first and second message digests are identical.

59. The interconnected system of claim 57, wherein verifying that ~~the~~ each entry in the plurality of entries within the update partition file has been generated by the server system includes signing the encrypted element associated with the entry with a public key of the server system, and the encrypted element of the update partition file has been prepared by signing, with the private key of the server system, a result of the application of an algorithm to data including a version of a setup password accessed by the server system.

60. The interconnected system of claim 59, wherein

the data includes the version of the setup password appended to ~~a~~ the entry,

said algorithm is a hash algorithm generating a message digest, and

verifying that the entry has been generated by the server system includes applying the hash algorithm to the setup password stored within the computing system appended the entry to generate a first version of a message digest and comparing the first version of the message digest with a second version of the message digest obtained by signing the encrypted element.

61. The interconnected system of claim 57, wherein

information stored in the protected partition is compared to each entry in the plurality of entries within the update partition file,

when a portion of the information stored in the protected partition is found to match the entry, the portion of the information stored in the protected partition is overwritten with the entry if space around the portion of the information stored in the protected partition is sufficient, and

when a portion of the information stored in the protected partition is not found to match the entry, the entry is appended to the information stored in the protected partition if space within the protected partition is sufficient.

62. The interconnected system of claim 61, wherein

the method additionally comprises receiving an input signal from a keyboard of the computing system and comparing the input signal with a signal corresponding to a setup password stored in non-volatile storage within the computing system, and

the protected partition is left unlocked if the input signal matches the signal corresponding to the setup password.

EVIDENCE APPENDIX

This appendix includes copies of the patent documents referenced herein.

RELATED PROCEEDINGS APPENDIX

None